

# FAST LABEL EMBEDDINGS FOR EXTREMELY LARGE OUTPUT SPACES

**Paul Mineiro & Nikos Karampatziakis**  
 Microsoft Cloud Information Services Lab  
 {pmineiro,nikosk}@microsoft.com

## ABSTRACT

Many modern multiclass and multilabel problems are characterized by increasingly large output spaces. For these problems, label embeddings have been shown to be a useful primitive that can improve computational and statistical efficiency. In this work we utilize a correspondence between rank constrained estimation and low dimensional label embeddings that uncovers a fast label embedding algorithm which works in both the multiclass and multilabel settings. The result is a randomized algorithm for partial least squares, whose running time is exponentially faster than naive algorithms. We demonstrate our techniques on two large-scale public datasets, from the Large Scale Hierarchical Text Challenge and the Open Directory Project, where we obtain state of the art results.

## 1 CONTRIBUTIONS

We provide a statistical motivation for label embedding by demonstrating that the optimal rank-constrained least squares estimator can be constructed from an optimal unconstrained estimator of an embedding of the labels. In other words, embedding can provide beneficial sample complexity reduction even if computational constraints are not binding.

We identify a natural object to define label similarity: the expected outer product of the conditional label probabilities. In particular, in conjunction with a low-rank constraint, this indicates two label embeddings are similar when their conditional probabilities are linearly dependent across the dataset. This unifies prior work utilizing the confusion matrix for multiclass (Bengio et al., 2010) and the empirical label covariance for multilabel (Tai & Lin, 2012).

We apply techniques from randomized linear algebra (Halko et al., 2011) to develop an efficient and scalable algorithm for constructing the embeddings, essentially via a novel randomized algorithm for partial least squares (Geladi & Kowalski, 1986). Intuitively, this technique implicitly decomposes the prediction matrix of a model which would be prohibitively expensive to form explicitly.

## 2 DESCRIPTION

### 2.1 NOTATION

We denote vectors by lowercase letters  $x, y$  etc. and matrices by uppercase letters  $W, Z$  etc. The input dimension is denoted by  $d$ , the output dimension by  $c$  and the embedding dimension by  $k$ . For an  $m \times n$  matrix  $X \in \mathbb{R}^{m \times n}$  we use  $\|X\|_F$  for its Frobenius norm,  $X^\dagger$  for the pseudoinverse,  $\Pi_{X,L}$  for the projection onto the left singular subspace of  $X$ .

### 2.2 PROPOSED ALGORITHM

Our proposal is Rembrandt, described in Algorithm 1. We use the top right singular space of  $\Pi_{X,L}Y$  as a label embedding, or equivalently, the top principal components of  $Y^\top \Pi_{X,L}Y$  (leveraging the fact that the projection is idempotent). Using randomized techniques, we can decompose this matrix without explicitly forming it, because we can compute the product of  $\Pi_{X,L}Y$  with another matrix  $Q$  via  $Y^\top \Pi_{X,L}YQ = Y^\top XZ^*$  where  $Z^* = \arg \min_{Z \in \mathbb{R}^{d \times (k+p)}} \|YQ - XZ\|_F^2$ . Algorithm 1

**Algorithm 1** Rembrandt: Response EMBedding via RANDomized Techniques

---

```

1: function REMBRANDT( $k, X \in \mathbb{R}^{n \times d}, Y \in \mathbb{R}^{n \times c}$ )
2:    $(p, q) \leftarrow (20, 1)$                                  $\triangleright$  These hyperparameters rarely need adjustment.
3:    $Q \leftarrow \text{randn}(c, k + p)$ 
4:   for  $i \in \{1, \dots, q\}$  do                                 $\triangleright$  Randomized range finder for  $Y^\top \Pi_{X,L} Y$ 
5:      $Z \leftarrow \arg \min \|YQ - XZ\|_F^2$ 
6:      $Q \leftarrow \text{orthogonalize}(Y^\top XZ)$ 
7:   end for                                          $\triangleright$  NB: total of  $(q + 1)$  data passes, including next line
8:    $F \leftarrow (Y^\top XQ)^\top (Y^\top XQ)$            $\triangleright F \in \mathbb{R}^{(k+p) \times (k+p)}$  is “small”
9:    $(V, \Sigma^2) \leftarrow \text{eig}(F, k)$ 
10:   $V \leftarrow QV$                                       $\triangleright V \in \mathbb{R}^{c \times k}$  is the embedding
11:  return  $(V, \Sigma)$ 
12: end function

```

---

is a specialization of randomized PCA to this particular form of the matrix multiplication operator. Fortunately, because  $X^\dagger(\Pi_{X,L} Y)_k$  is the optimal rank-constrained least squares weight matrix, Rembrandt is a randomized algorithm for partial least squares (Geladi & Kowalski, 1986).

Algorithm 1 is inexpensive to compute. The matrix vector product  $YQ$  is a sparse matrix-vector product so complexity  $O(nsk)$  depends only on the average (label) sparsity per example  $s$  and the embedding dimension  $k$ , and is independent of the number of classes  $c$ . The fit is done in the embedding space and therefore is independent of the number of classes  $c$ , and the outer product with the predicted embedding is again a sparse product with complexity  $O(nsk)$ . The orthogonalization step is  $O(ck^2)$ , but this is amortized over the data set and essentially irrelevant as long as  $n > c$ . Furthermore random projection theory suggests  $k$  should grow only logarithmically with  $c$ .

### 2.3 EXPERIMENTS

Table 1: Data sets used for experimentation and times to compute an embedding. Timings are for a Matlab implementation on a standard desktop (dual 3.2Ghz Xeon E5-1650 CPU and 48Gb of RAM).

Dataset	Type	Modality	Examples	Features	Classes	Rembrandt $k$	Time (sec)
ODP	Multiclass	Text	$\sim 1.5M$	$\sim 0.5M$	$\sim 100K$	300	6,530
LSHTC	Multilabel	Text	$\sim 2.4M$	$\sim 1.6M$	$\sim 325K$	500	8,006

Table 2: ODP results.  $k = 300$  for all embedding strategies. RE = Rembrandt; CS = compressed sensing; PCA = unsupervised (feature) embedding; LT = LomTree (Choromanska & Langford, 2014); “A+LR” = logistic regression on representation A.

Method	RE + LR	CS + LR	PCA + LR	LT
Test Error	83.15%	85.14%	90.37%	93.46%

Table 3: LSHTC results. FastXML and LPSR-NB are from Prabhu & Varma (2014). “A+ILR” = independent logistic regression on representation A.

Method	RE ( $k = 800$ ) + ILR	RE ( $k = 500$ ) + ILR	FastXML	LPSR-NB
Precision-at-1	53.39%	52.84%	49.78%	27.91%

## REFERENCES

- Bengio, Samy, Weston, Jason, and Grangier, David. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, pp. 163–171, 2010.
- Choromanska, Anna and Langford, John. Logarithmic time online multiclass prediction. *arXiv preprint arXiv:1406.1822*, 2014.

- Geladi, Paul and Kowalski, Bruce R. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.
- Halko, Nathan, Martinsson, Per-Gunnar, and Tropp, Joel A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- Prabhu, Yashoteja and Varma, Manik. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 263–272. ACM, 2014.
- Tai, Farbound and Lin, Hsuan-Tien. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.